Component based software – the obligatory practical study

Lars-Åke Fredlund

The obligatory practical study: tasks

The practical study is **individual**; no working in groups

The obligatory practical study: tasks

The practical study is **individual**; no working in groups

- **Select** a task concerning component-based software that interests *you*

The obligatory practical study: tasks

The practical study is **individual**; no working in groups

- **Select** a task concerning component-based software that interests *you*
- **Email** me (`lfredlund@fi.upm.es`) a description of the task to get an ok (**now**)

The obligatory practical study: tasks

The practical study is **individual**; no working in groups

- **Select** a task concerning component-based software that interests *you*
- **Email** me (`lfredlund@fi.upm.es`) a description of the task to get an ok (**now**)
- Start working (**begin now**)

The obligatory practical study: tasks

The practical study is **individual**; no working in groups

- **Select** a task concerning component-based software that interests *you*
- **Email** me (`lfredlund@fi.upm.es`) a description of the task to get an ok (**now**)
- Start working (**begin now**)
- Give a presentation (around 25 minutes)
  (**Starting January 13, 2010; 4 persons/day**)

The obligatory practical study: tasks

The practical study is **individual**; no working in groups

- **Select** a task concerning component-based software that interests *you*
- **Email** me (`lfredlund@fi.upm.es`) a description of the task to get an ok (**now**)
- Start working (**begin now**)
- Give a presentation (around 25 minutes)
  (**Starting January 13, 2010; 4 persons/day**)
- A report (15–20 pages) – Spanish allowed
  (**Mid February 2010**)

The obligatory practical study: tasks

The practical study is **individual**; no working in groups

- **Select** a task concerning component-based software that interests *you*
- **Email** me (`lfredlund@fi.upm.es`) a description of the task to get an ok (**now**)
- Start working (**begin now**)
- Give a presentation (around 25 minutes)
  (**Starting January 13, 2010; 4 persons/day**)
- A report (15–20 pages) – Spanish allowed
  (**Mid February 2010**)
- **Participate** (ask questions) at other students' presentations

What can be a study topic

Abstract examples:

- Study and use one of the component frameworks,
- `or` specify, implement, and validate a set of components,
- `or` study the impact and/or problems (economic, timewise) of introducing components in software development,
- `or` study and use software architecture description methods,
- `or` study and use a method for specifying formally component interfaces with regards to component *behaviour*

Concrete task examples

- *Development of a Business Process using Microsoft BizTalk server*
- or *Modelling and analysis of a highly concurrent auction system using the π calculus process algebra*
- or *Studying the consequences of the introduction of a component-based development methodology at company XXX*
- See course web page, course notes, wikipedia, google, . . . for more inspiration

A mindset for doing the practical study

- You work in a group at company XXX doing software development

A mindset for doing the practical study

- You work in a group at company XXX doing software development
- Your group manager wants you to evaluate a new development method/software tool centered around component based thinking that might be interesting for the group to start using

A mindset for doing the practical study

- You work in a group at company XXX doing software development
- Your group manager wants you to evaluate a new development method/software tool centered around component based thinking that might be interesting for the group to start using
- How do you do the evaluation?
  Well you read about it. . .

A mindset for doing the practical study

- You work in a group at company XXX doing software development
- Your group manager wants you to evaluate a new development method/software tool centered around component based thinking that might be interesting for the group to start using
- How do you do the evaluation?
  Well you read about it. . . But you don’t trust all the hype you read so you apply it to a **serious case study**, analysing the tool for **benefits/weaknesses**

A mindset for doing the practical study

- You work in a group at company XXX doing software development
- Your group manager wants you to evaluate a new development method/software tool centered around component based thinking that might be interesting for the group to start using
- How do you do the evaluation?
  Well you read about it. . . But you don't trust all the hype you read so you apply it to a **serious case study**, analysing the tool for **benefits/weaknesses**
- Later on you report your findings to the group (and your manager) in a short seminar, and in a written report summarizing your findings regarding the tool:
  - what is about?
  - is it any good? (giving reasons based in your case study!)
  - should your group start using the tool?

About the practical study

It is **not** just a literature study; I do not want to read 12 pages of an introduction to Web Services extracted from Wikipedia

- **Learn** a framework
- **Apply** the framework to a *challenging example*, as part of a critical **Evaluation**

  Program a solution, write a specification and test, use an architecture description language to specify an architecture…
- Example: *travel agency* (2008), *auction system* (2007), or your own choice
- **Document** the **result** of applying the framework to the example, with criticism resulting from **your** study: *did things work?*, *what were the benefits compared to not using the framework?*, *what were the problems?*, etc
- **Do not be afraid to include concrete details in the report:** source code, specifications, etc (perhaps as appendices).

An “example” example: The Travel Agency

The travel agency sells ”travel packages” to customers. A travel package contains a set of travel items, which can typically be:

- airplane tickets, train tickets, bus tickets requests
- car rental requests
- hotel room requests
- travel insurance request

An “example” example: The Travel Agency

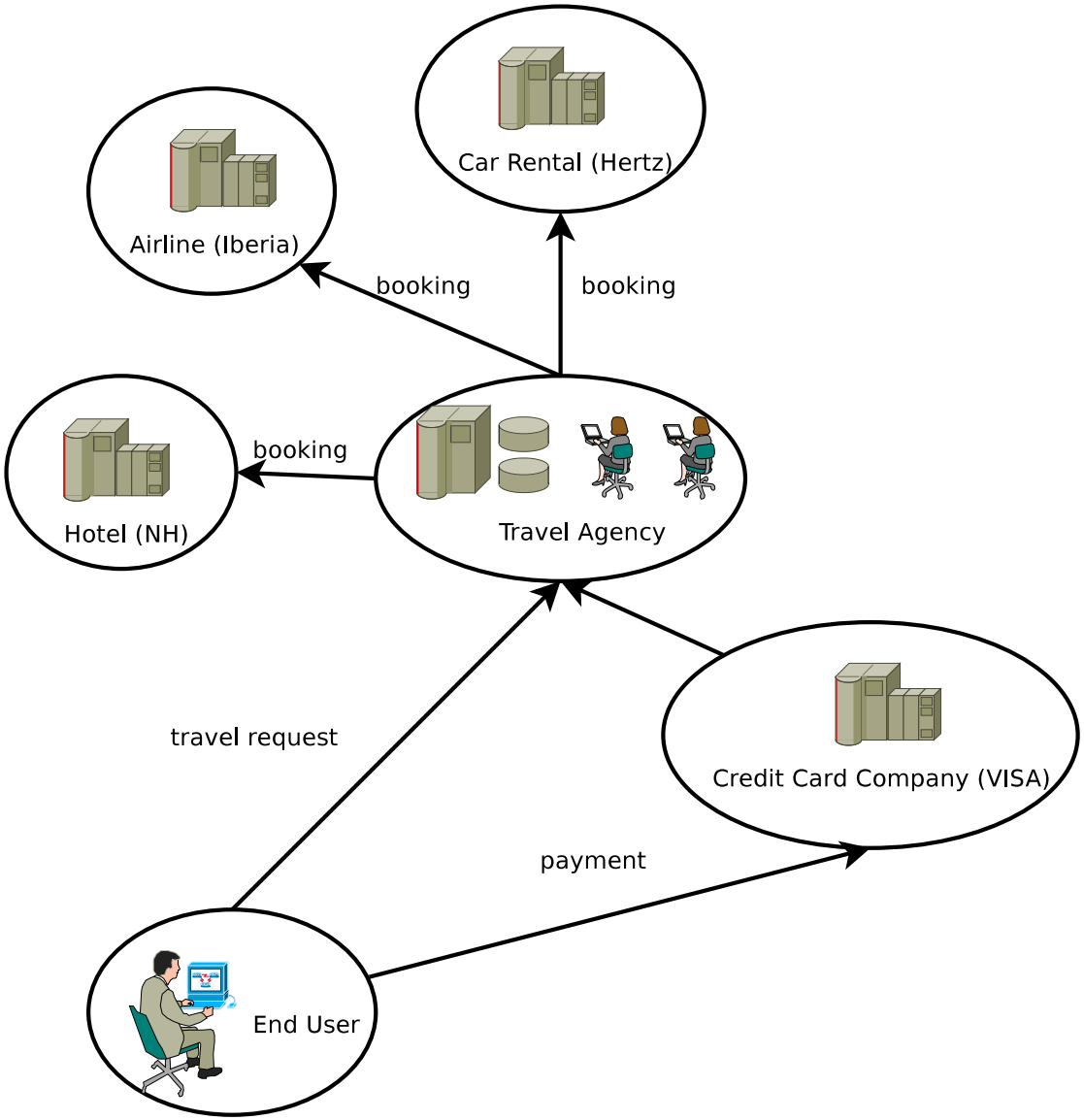
The travel agency sells ”travel packages” to customers. A travel package contains a set of travel items, which can typically be:

- airplane tickets, train tickets, bus tickets requests
- car rental requests
- hotel room requests
- travel insurance request

A user of the travel agency submit booking requests for travel packages; such by necessity have only partial information.

- For example: requests for airplane tickets typically have no airline specified, time information is partial (”wish to travel in the morning”)

Travel Agency Example



Travel Agency Example: Interactions

- A customer submits a request for a travel package (typically containing several travel items)
- The system searches (by interacting with the web services of relevant companies) for solutions to the travel package
- If there are no solutions the user is invited to reformulate his request for a travel package
- If the user chooses a package the system collects billing information (credit card), and attempts to get payment, and in the same *transaction* also confirms and pays the separate travel items of the travel package
- Problem are reported to the user – if any part fails, the whole transaction should fail

The example can be extended to treat additional themes such as timing deadlines, interfaces, etc.

Travel Agency Study Concerns

Concurrency: ensure that your system can copy with multiple booking attempts simultaneously; talk to multiple travel item providers concurrently (Iberia, Air France,. . .)

Timing: don’t wait indefinitely for answers from providers of travel items (e.g., ”if Iberia doesn’t answer within 30 seconds, provide the possible solutions from the other airlines”)

Management: failed bookings should be reported, both to the user but also on a **”management interface”** for travel agency internal use, so that manual assistance can be provided to customers having difficulties. Moreover, the management interface should provide information about current activities (e.g., ”how many ongoing bookings are there right now”) to aid the travel agency to optimise its resources

Some practical study ideas from today (Nov 25th 2009)

- Use the **pi4soa** tool (based in WS-CDL) to design an auction system (see specification of an auction on the web page)
- Evaluate the **Spec#** or the **Code Contracts Tools** from Microsoft Research for contract-based programming (you need access to Visual Studio)
- Evaluate the **QuickCheck/Haskell** or **QuickCheck/Java** or **QuickCheck/Erlang** tools for property-based development (and random testing/test driven development)
- Study one of the mashup platforms (for example **EzWeb**: `http://ezweb.tid.es`)
  - What does it offer in terms of a component platform?
  - What are the communication mechanisms?
  - Develop an example to evaluate the platform